

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ
Государственное бюджетное общеобразовательное учреждение города Москвы
«Школа № 1571» (ГБОУ Школа №1571)
ул. Фрунзенский д. 1, к. 1, Москва, 125141 телефон (495) 492-35-71, тел. (495) 492-35-11
ул. Пискаревский д. 12, к. 2, Москва, 125481 тел. (495) 495-1530, (495) 497-62-96;
ИНН 770326624 КПП 77031001 ОГРН 105773932776 ОКПО 53817110
E-mail: 1571@obrn.mos.ru; <http://www.ko1571sz.mosobrn.ru>

Университет Правительства Москвы
У конкурса исследовательских и проектных работ обучающихся
образовательных организаций города Москвы и Московской области
«Мегалитис XXI века – город для жизни» в 2020/2021 учебном году

Конкурсная работа

На тему: « Приложения для воспроизведения нот музыкальных инструментов в
образовательных целях»

Выполнила: Шаповалова И.А.,

Ученица 10 «Д» класса

Научный руководитель:

Ст. преп. каф. АПид НИТУ МИСиС

Зорин И.А., учитель Демидова О.В.

Подпись

Директор ГБОУ Школа №1571:
Барганял Марина Викторовна

ГБОУ

Подпись

№ 1571

Москва
2021 г.

Проект Шаповаловой Ирины Алексеевны «Приложение для воспроизведения нот музыкальных инструментов в образовательных целях» представляет собой приложение для ОС «Android» и выполнен на языке Java. Цель создания приложения состоит в том, чтобы воспроизводить мелодию по переданным на вход нотам звучанием выбранного пользователем инструмента.

В настоящий момент аналоги данного приложения в Google Play отсутствуют, что говорит о высокой степени актуальности продукта. Данное программное решение может быть полезно музыкантам, в том числе учащимся музыкальных учебных заведений. Работа выполнена Ириной Алексеевной самостоятельно на высоком профессиональном уровне и заслуживает наивысшей оценки.

Ст. преп. каф. АПид НИТУ МИСиС



Зорин И.А.

Оглавление

| | |
|--|----|
| Введение..... | 4 |
| Цели и задачи | 4 |
| 1. Методика выполнения работы..... | 5 |
| 1.1. Дорожная карта проекта..... | 5 |
| 1.2. Анализ существующих решений..... | 5 |
| 1.3. Используемое оборудование и программное обеспечение | 6 |
| 1.4. Принцип работы приложения..... | 6 |
| 1.5. Варианты работы программного кода. | 7 |
| 2. Заключение | 13 |
| 3. Список используемой литературы и интернет источников | 13 |
| Фото и видео материалы: | 14 |

Введение

У людей с ограниченными возможностями, увлекающихся музыкой, зачастую отсутствует возможность заниматься с педагогом или возникают трудности в частом посещении образовательных организаций и репетиций. Из-за этого у них могут отсутствовать музыкальные навыки или нужные музыкальные инструменты, возникают сложности в игре в оркестре, необходимы более частые занятия и репетиции с педагогом и оркестром. Им необходимо прослушать, как должны звучать ноты, которые у них в данный момент на руках (зачастую в бумажном виде или текстовом файле). Специализированное приложение смогло бы компенсировать эти затруднения, помочь отработать и автоматизировать навыки.

Цели и задачи

Цель:

Создание приложения, с помощью которого можно загрузить файл с нотами формата PNG, PDF или JPEG (возможно и другие форматы), выбрать один или несколько музыкальных инструментов и проиграть ноты для репетиции, самостоятельных занятий и аккомпанемента.

Задачи:

1. Изучить, как загрузить файл с нотами с помощью языка Java.
2. Преобразовать программно текстовое обозначение нот в звук.
3. Настроить программно длительность нот, их громкость.
4. Узнать и применить, как воспроизвести ноту нужной октавы.
5. Изучить и применить, как программно указать и использовать необходимые музыкальные инструменты для воспроизведения файла.

6. Применить на практике и углубить свои знания в области языка Java.
7. Установить необходимое программное обеспечение Java.
8. По возможности сформировать мобильное приложение для Android.

1. Методика выполнения работы

1.1. Дорожная карта проекта

| Направление работы, ключевые задачи, Сроки | Сентябрь | Октябрь | Ноябрь | Декабрь | Январь |
|--|----------|---------|--------|---------|--------|
| Определение с темой проекта | | | | | |
| Определение с языком программирования и платформами разработки | | | | | |
| Создание программного кода | | | | | |
| Разработка интерфейса | | | | | |
| Подготовка конференции | | | | | |

1.2. Анализ существующих решений

На данный момент времени существует два немного похожих приложений. Первое это Guitar Pro. В этом приложении есть существенные минусы, а именно: ручной ввод нот, применяется только для компьютера, нельзя выбрать более двух инструментов для проигрывания музыкальных

произведений. А второе это типичный музыкальный симулятор. К сожалению, в нем нельзя внести ноты и требуются музыкальные навыки.

1.3. Используемое оборудование и программное обеспечение

- персональный компьютер
- операционная система Windows10
- среда разработки на языке Java Eclipse
- среда разработки Android Studio
- виртуальное устройство Android Virtual Device
- интернет

1.4. Принцип работы приложения

Приложение можно сформировать для различных типов устройств смартфон, планшет, компьютер. Общий вид представлен на рис.1.

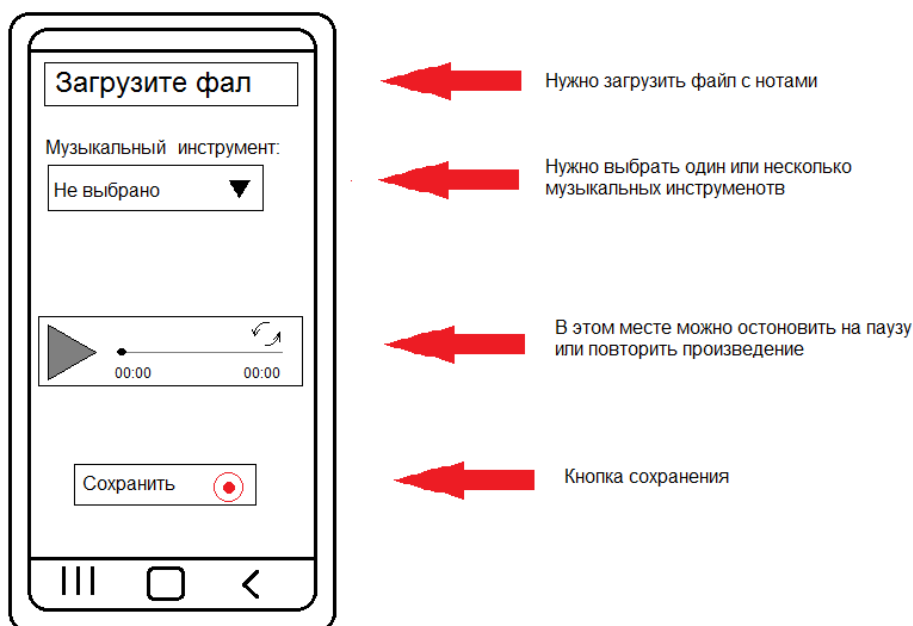


Рис.1 Внешний вид приложения

По кнопке «Загрузить файл» пользователь выбирает файл с нотами на своём устройстве. Далее в поле «Музыкальный инструмент», пользователь может выбрать один или более музыкальных инструментов в зависимости от музыкального произведения. По кнопке воспроизведения можно прослушать или остановить музыку. По кнопке «Сохранить» можно сохранить звуковой файл.

1.5. Варианты работы программного кода.

I. Вариант воспроизведения текстового файла.

Предполагается, что в текстовом файле содержатся данные о канале, длительности, громкости, номере инструмента, номере ноты в строковом формате.

На рис. 2 представлена часть программного кода, в котором я указываю все библиотеки для работы со звуком и текстовым файлом.

```
package FluteDilet;

import java.util.logging.Level;
import java.util.logging.Logger;
import javax.sound.midi.MidiChannel;
import javax.sound.midi.MidiSystem;
import javax.sound.midi.MidiUnavailableException;
import javax.sound.midi.Synthesizer;
//библиотеки для работы с файлами
import java.io.File;
import java.util.*;

//подключаем библиотеки
```

Рис.2 Подключение библиотек.

Далее представлен кусок программного кода (рис. 3), где я подключаю класс плеера, который воспроизведет нужный нам звук по параметрам: канал, длительность, громкость, номер инструмента, номер ноты.

```

public class FluteDilet6 {

    private MidiChannel[] channels = null;
    private Synthesizer synth = null;

    public FluteDilet6() {
        try {
            synth = MidiSystem.getSynthesizer();
            synth.open();
            channels = synth.getChannels();
        } catch (MidiUnavailableException ex) {
            Logger.getLogger(FluteDilet6.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void close() {
        synth.close();
    }

    public void playSound(int channel, int duration, int volume, int instr, int... notes)
    {
        //Кладутся параметры канал, длительность, громкость, номер инструмента, номер
        ноты
        for (int note : notes) {
            channels[channel].programChange(instr);
            channels[channel].noteOn(note, volume);
        }
        try {
            Thread.sleep(duration);
        } catch (InterruptedException ex) {
            Logger.getLogger(FluteDilet6.class.getName()).log(Level.SEVERE, null, ex);
        }
        for (int note : notes) {
            channels[channel].noteOff(note);
        }
    }
}

```

Рис. 3 Подключение класса плеера

На рис.4.1 и рис.4.2 представлен код, который открывает, читает и записывает данные из файла в двумерный массив в числовом виде.

Все данные из файла помещаются в динамический список в строковом виде, из этого списка мы создаем двумерный массив с параметрами типа «число».


```

public static void main(String[] args) throws MidiUnavailableException, Exception
{
    //получи необходимые нам данные из текстового файла

    // передать путь к файлу в качестве параметра
File file = new File("C:\\Проект\\001.txt");

try (Scanner sc = new Scanner(file)) {
    char a = ' ';
    String str = null;
    String gromstr = "";
    String kanalstr = "";
    String instr = "";
    String notastr = "";
    String dlitstr = "";

    String delimiter = "";

    int [][] notes;
    int SCh = 0;
    List<String[]> A = new ArrayList<String[]>();

    while (sc.hasNextLine()) {

        //SCh++;

        str = String.valueOf(sc.nextLine());
        a = str.charAt(0);
        if(a=='g') {
            for(int i=1; i<= str.length()-1; i++) {
                gromstr = gromstr +
String.valueOf(str.charAt(i));
            }
        }
        if(a=='k') {
            for(int i=1; i<= str.length()-1; i++) {
                kanalstr = kanalstr +
String.valueOf(str.charAt(i));
            }
        }
        if(a=='i') {
            for(int i=1; i<= str.length()-1; i++) {
                instr = instr + String.valueOf(str.charAt(i));
            }
        }
        if(a == '0' || a == '1' || a == '2' || a == '3' || a == '4' || a
== '5' || a == '6' || a == '7' || a == '8' || a == '9' ) {
            delimiter = "\\,"; // Разделитель

```

Рис.4.1

```

разделённой строки
String[] subStr = str.split(delimeter); //создание
// Вывод результата
for(int ii = 0; ii < subStr.length; ii++) {
    //System.out.println(subStr[ii]);
    if (ii == 0)
    {
        dlitstr = subStr[ii];
    }
    else
    {
        notastr = subStr[ii];
    }
}
A.add(new String[]
{kanalstr,dlitstr,gromstr,instr,notastr});
}

for (String[] row : A) {
    SCh++;
}

notes = new int[SCh][];
SCh = 0;
for (String[] row : A) {
    notes[SCh] = new int[]{Integer.parseInt(row[0]),
Integer.parseInt(row[1]), Integer.parseInt(row[2]), Integer.parseInt(row[3]),
Integer.parseInt(row[4].trim())};
    SCh++;
}

```

Рис.4.2

Далее с помощью плеера проигрываются строки массива, каждая из которых состоит из 5 параметров: канал, длительность, громкость, номер инструмента, номер ноты. Одна строка массива представляет собой один звук произведения (рис.5).

```

        FluteDilet6 player = new FluteDilet6();

        for (int[] note : notes) {
            if (note[4] != -1) {
                player.playSound(note[0], note[1], note[2],note[3],
note[4]);
            }
            else {
                try {
                    Thread.sleep(note[1]);
                } catch (InterruptedException ex) {
                    Logger.getLogger(FluteDilet6.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        }

        player.close();
    }
}

```

Рис.5

II. Вариант воспроизведения звукового файла.

Предполагается, что пользователь загрузил готовый звуковой файл форматов WAVE, MIDI.

Средства Java позволяют воспроизводить звуковые файлы без использования внешних библиотек. Поддерживаются моно и стереозвук с частотой от 8 до 48 кГц.

```
package kkk;
```

```
import java.io.File;
import java.io.IOException;

import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;

public class kkk {
    public static void main(String[] args)
    {
        try {

            File soundFile = new File("c:\\Proekt\\Bach Johann Sebastian.mid"); //Звуковой файл Bach Johann Sebastian
            //Получаем AudioInputStream

            AudioInputStream ais = AudioSystem.getAudioInputStream(soundFile);

            //Получаем реализацию интерфейса Clip
            Clip clip = AudioSystem.getClip();

            //Загружаем наш звуковой поток в Clip
            clip.open(ais);

            clip setFramePosition(0);
            clip.start();

            Thread.sleep(clip.getMicrosecondLength()/1000);
            clip.stop();
            clip.close();

        } catch (IOException | UnsupportedAudioFileException | LineUnavailableException exc) {
            exc.printStackTrace();
        } catch (InterruptedException exs) {}

    }
}
```

Рис.6

Существуют два основных способа воспроизведения звука:

- использование интерфейса Clip
- использование интерфейса SurceDataLine

Различие между ними состоит в том, что первый способ подразумевает загрузку воспроизводимого файла в память целиком. Это удобно при воспроизведении небольших файлов или неоднократном проигрывании.

При использовании второго способа файл загружается в память частями в буфер, размер которого можно установить произвольно. На рис. 6 реализован первый способ воспроизведения звука.

2. Заключение

Созданное приложение поможет людям с ограниченными возможностями в занятиях музыкой, расширит их возможности, поможет отработать и автоматизировать навыки игры на любом музыкальном инструменте, игры в оркестре или под аккомпанемент без очного присутствия педагога. Приложение можно использовать, находясь в любом месте с использованием и без использования интернета.

3. Список используемой литературы и интернет источников

1. 1С: Учебный центр №1. Основы программирования на языке «Java» для школьников. Методические материалы для слушателя сертифицированного курса. Модуль 1. – М: 2013
2. 1С: Учебный центр №1. Основы программирования на языке «Java» для школьников. Методические материалы для слушателя сертифицированного курса. Модуль 2. – М: 2013
3. 1С: Учебный центр №1. Основы программирования на языке «Java» для школьников. Методические материалы для слушателя сертифицированного курса. Модуль 3. – М: 2013

4. 1С: Учебный центр №1. Основы программирования на языке «Java» для школьников. Методические материалы для слушателя сертифицированного курса. Модуль 4. – М: 2013
5. Java | Java и Вы, загрузите сегодня [электронный ресурс] - <https://www.java.com/ru/> - Заглавие экрана. – (Дата обращения: 06.10.2020)
6. Eclipse foundation [электронный ресурс] - <https://www.eclipse.org/> - Заглавие экрана. – (Дата обращения: 30.09.2020)
7. Android Studio [электронный ресурс] - <https://developer.android.com/studio?hl=ru> - Заглавие экрана. – (Дата обращения: 10.11.2020)
8. Эмулятор Android Studio SDK [электронный ресурс] - <https://android-sdk.ru/> - Заглавие экрана. – (Дата обращения: 03.11.2020)

Фото и видео материалы:

<https://cloud.mail.ru/public/QVeL/VtL2Vb6cL>